NAS9-12402-061
DRL L.I. 6, T-714

NASA CR-
144378

(NASA-CR-144378) SHUTTLE CHECKOUT SOFTWARE SYSTEM APPLICATION VERIFICATION LABORATORY MANUAL Final Report (General Electric Co.) 16 p HC $3.25 CSCL 22B G3/19 — N75-29169 Unclas 32402

SHUTTLE CHECKOUT SOFTWARE SYSTEM

APPLICATION VERIFICATION LABORATORY MANUAL

FINAL REPORT

JULY 1975

This document was prepared for the National Aeronautics and Space Administration Lyndon B. Johnson Space Center under Contract NAS9-12402, Data Requirement List, Line Item 6, DRD MA-129T.

General Electric Company
Space Division
Technical and Support Services
Houston Operations

A.V.L. USERS MANUAL

REFERENCE DOCUMENTS

UTE Test Language Users Manuel UTE0.5.0.0.7.0

AVLS Software Users Manual UTE0.1.3.0.7.25

SAIL Test Operations Center Users Guide UTE0.5.0.0.7.7

1.0 INTRODUCTION AND PURPOSE

1.1 INTRODUCTION

The Applications Verification Laboratory (AVL) will provide both the hardware and software to allow closed loop verification of UTE Test Language Programs. The AVL can be divided, for discussion, into two basic groups or functions which are the test equipment simulator and the UTE System. (Ref. Figure 1. AVL Block Diagram).

The Simulator (AVLS) allows the user to generate interactive simulations of test article functions necessary to exercise UTETL programs. The AVLS is controlled by an operating system program executing in conjunction with a simulation language program. The AVLS will produce a PCM data stream destined for the UTE system. The AVLS will receive commands from the UTE and calculate new values for insertion into the PCM stream, as directed by the simulation program.

The UTE system is basically a single console DCM with all the necessary peripheral equipment required for operational test. This system will allow the UTETL program under test to be evaluated in an operational environment similar to the SAIL TOC. The UTE will receive PCM data from the simulator and output control commands as directed by the UTETL program.

The interaction of the AVLS and the UTE system allows the checkout engineer to exercise (debug & verify) the UTETL applications programs to reach a level of confidence prior to applying the test program to an actual test article.

1.2 PURPOSE

This document is intended to familiarize the Applications Verification Laboratory (AVL) user with the basic capabilities of the system as well as the procedure involved in utilizing the AVL to test a UTETL program. In general, detailed hardware and software descriptions will not be included as they are available in referenced documents.

2.0 AVL HARDWARE CONFIGURATION

The AVL is composed of the hardware listed in Table 1 and configured per Figure 1. As the UTE and common area hardware elements are described in the SAIL TOC Users Guide (UTE 0.5.0.0.7.7)., they will not be covered herein.

2.1 AVLS HARDWARE ELEMENTS

2.1.1 META 4 SYSTEM

The purpose of this system is to execute the test equipment simulation language program and provide that program with incoming commands, an operator interface, and data output capability. It consists of a META-4 Central Processor, 3K of ROM, 48K of random access core memory, a real time clock, I/C memory, and an extended operators panel.

2.1.2 DISPLAY CABINET

The simulator/operator interface is via a single DCM type data monitor CRT, VF plasma display, and a DCM III type keyboard. Interface of the keyboard and displays with the META 4 is standard UTE except that trend is not required . A refresh memory/display processor interface is not required as the META 4 will perform display processing.

2.1.3 PERIPHERAL CABINET

This cabinet contains both the moving head disk unit and the magnetic tape unit controller.

The moving head disk is a UTE type and functions as the system disk for either the 1130 disk monitor system for the program preparation mode or the simulation operating system for the simulation mode. The mag tape controller provides the control interface between the META 4 and the associated 606 digital tape unit.

2.1.4 COMMUNICATIONS CONTROLLER

The Communications Controller provides the primary link between the AVLS and the UTE. It acquires commands from the checkout system, performs command reformatting and identification for the simulation controller, and outputs the commutated data to the PCM generator.

3.0 TESL ORGANIZATION AND USE

The Test Equipment Simulation Language (TESL) is a derivative of the Unified Test Equipment Test Language (UTETL), specifically modified for use in the Applications Verification Laboratory (AVL). As the languages are closely related both in format and use, this document will rely heavily on the UTETL Users Manual (UTE 0.5.0.0.7.0) for reference to common material.

3.1 CAPABILITIES

The TESL allows the user to generate interactive simulations of test article functions necessary to exercise UTETL programs. The TESL utilizes the UTETL Syntax and provides, essentially, the same monitor and control capabilities. The TESL receives control commands via the DBT

and outputs response data via the PCM stream. The generation of this PCM stream is an automatic function and does not require supervision from the simulation program, however, PCM transmission can be turned on and off by the TESL program.

3.2 PROGRAM ELEMENTS

A TESL program may contain four unique sections which are functionally identical to the sections of a UTETL program. The TESL sections are as follows:

$SPECIFICATION -

This section is required and only one section of this type is allowed per program. The specification provides the correlation between T.L. names and hardware response and stimulus points being simulated.

$PROCEDURE -

This section is required and only one section of this type is allowed per program. The procedure defines the AVLS control functions which sequence the execution of taps and control the run-time operator interface. The procedure section is analogous to the "Main line program" in other languages.

$TAPS -

This section is optional and a program may contain more than one of this type section. $Taps define the operational processes which calculate the outputs of the devices being stimulated. This section corresponds to the $Tests section in UTETL.

$DISPLAY -

This section is optional and a program may contain more than one of this type section. The display section defines the background, data content and display formats of the information presented to the operator. Unlike UTETL, there is only one CRT display present in the AVLS.

3.3 TESL SECTION STATEMENTS

The majority of all TESL statements are identical to UTETL statements in both syntax and use. There are, however, some new statements as well as some modification in use of existing statements. As the UTETL statements are described in detail in the UTETL Users Manual (UTE 0.5.0.0.7.0), only the modifications will be detailed herein.

3.3.1 SPECIFICATION STATEMENTS

The TESL statements and their functional use for this section are identical to those of UTETL with the following exceptions:

a. MEASUREMENTS defined in this section are actually output data destined for the PCM stream.

b. STIMULI defined in this section are actually input data to the simulator.

c. GLOBALS are not allowed in the TESL.

d. The sync variable specified by DEFINE SYNC 512 or DEFINE SYNC 128 is used to control the on/off state of the PCM stream.

e. TRANSMIT is a new statement in the specification section which allows the user to simulate test equipment communication on the comm. bus. The transmit statement will select the variable for transmission to the Test Operating System (TOS) via the

communications bus and will define the format (raw counts or engineering units) for real measurements. Pseudo measurements will be formatted as engineering units. The transmission will be activated whenever the associated named variable is the target of a replace statement in a procedure or a taps section. Refer to figure 2, for an illustration of this statement.

f. DEFINE LDB is a new statement used to assign a name to the system variable which will control the mode (powered-unpowered) of the launch data bus interface of the AVLS with the TOS. This statement utilizes the same general format as define time.

g. DOWNLIST is a new statement which will select a variable for transmission to the TOS via the asynchronous technique imbedded in the PCM stream. This function is not currently implemented in the AVLS software and is mentioned here for information only.

3.3.2 $PROCEDURE STATEMENTS

The TESL statements for this section are identical to those of UTETL with the following exceptions:

a. SET statements are not allowed in the TESL.

b. MEAS statements are not allowed in the TESL.

c. STIMULI, as defined in the specification section, may be used in this section as real inputs to the simulator. The same rules which apply to real measurements in UTETL, apply to stimuli in TESL.

d. REAL MEASUREMENTS in TESL are treated in the same manner as pseudos in UTETL.

3.3.3 $TAPS STATEMENTS

The TESL statements for this section are identical to those of the $TESTS section of UTETL with the following exceptions:

a. SET statements are not allowed in the TESL.

b. TEST is replaced by taps.

c. The rules for stimuli and real measurements used in TESL procedure also apply in this section.

3.3.4 $DISPLAY STATEMENTS

The TESL statements for this section are identical to those of UTETL with the following exceptions:

a. PAGEC statement is not used in TESL.

b. The rules for generating a display page in TESL are the same as those in UTETL for a data page except that line (0) and lines (22, 23, 24) are reserved for system use.

3.4 TESL USE

The TESL utilization may be broken into two basic phases for discussion. The program preparation phase and the operational phase are similar to the procedures described in sections 4.6 and 4.7 of the SAIL TOC users guide (UTE 0.5.0.0.7.7), however, there are some differences due to dissimilar hardware elements.

3.4.1 PROGRAM PREPARATION

As the TESL compiler is a modified version of the UTETL compiler, the program preparation sequence is identical to that of UTETL with the following exceptions:

a. The TESL compiler may be run in eitner the simulator or a DCM.

b. Compilation of the $SPECIFICATION section will result in two binary files being generated. These two files will be headed by hollerith name cards as follows:

COL1 COL800

// _ NAME _ $$MDL$$ MDL00000

// _ NAME _ $$PCMAP$$ COM00000

c. The TESL compiler does not output a "NO ERRORS" message to the line-printer upon successful compilation of a SSPECIFICATION section.

3.4.2 OPERATIONAL USE

In operation, the TESL program will appear functionally identical to a UTETL program with the following exceptions:

a. There is no fixed head disk in the simulator, therefore, all programs are stored on the moving head disk and called off as required.

b. The AVL operating software does not require a password to call up procedures.

c. In the system loading sequence the operating system will display a mode select page and corresponding VFK's. The SIMULATE mode available here corresponds to the TEST mode in the UTE system.

d. It should be noted that the simulator operating system processes test language parameters in engineering units up to the time they are passed to the PCM generator. At this time they are converted into counts and must be reconverted

at the receiving DCM. This process will result in some disparity between the data actually calculated and displayed at the simulator and the resulting data at the DCM. This delta should not exceed $\pm 1\%$ of full scale, however, it could cause some unexpected results if not anticipated.

4.0 AVL HARDWARE AND SOFTWARE INTERACTION

In order to better understand the hardware/software interaction, this section will attempt to outline a closed loop data interchange in a step-by-step fashion. This outline will assume that both the simulator and the UTE system are initialized with test language programs in operation. It is also assumed that the test operator will initiate the action from the DCM. The closed loop sequence of events could be as follows:

1. Test operator selects a VFK and XEQ. -

   This action will cause the BOSS to commence sequential processing of the T.L. procedure statements at the indicated entry point.

2. Command Transmission -

   Assuming that one of the sequential statements in this block is a set statement, the BOSS/TOS will route the command to the ACM where it will be reformatted and transmitted to the simulator via the data bus.

3. Execute Simulation Response -

   The simulator will receive this command and pass it to the communications controller for reformatting. The META 4 will treat the reformatted command from the CCU as a real measurement input.

As the simulation program recognizes a change in this command, it will perform the necessary data manipulations to simulate the appropriate hardware responses.

4. Data Output -

   As data is modified by the simulation program, the new values are stored in the master data list by the simulation operating software (SOS). The CCU will be performing a continuous PCM commutation function and at the proper timeslot during this function the commutator program will fetch the value of that data from the master data list. This data will be placed in the PCM output register. From here, the data will be inserted into the PCM stream by the PCM generator.

5. Data Input -

   The PCM stream will be received and processed by the acquisition and command module (ACM). The ACM will decommutate and perform initial data compression on the PCM data. That data which has changed enough to pass through the limit checker will be formatted and routed to the DCM for processing by the test language program.

6. Data Utilization -

   The DCM will receive the formatted data from the ACM and process it as directed by the test language program. This will close the loop of testing from command initiation to simulation response.

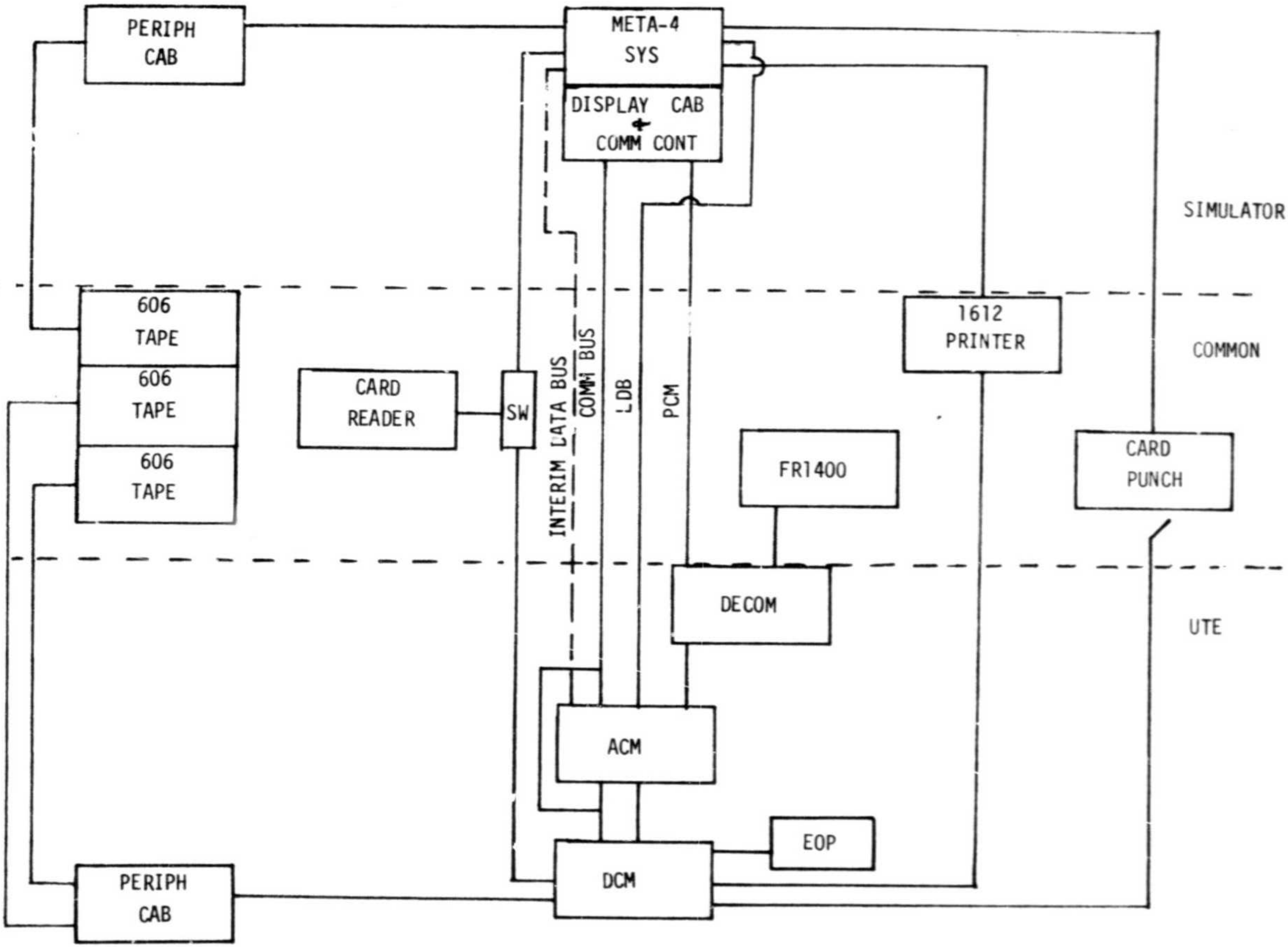

FIGURE 1 AVL CONFIGURATION

GENERAL ELECTRIC

FORTRAN CODING FORM

PROBLEM: FIGURE 2 - TRANSMIT ILLUSTRATION

CODER | DATE | PAGE

```
$SPECIFICATION (ILLUS), (IOB)
    2  DEFINE ANA  XAI(SP1001U), XA3(SP4139U)
    5  STATES = LO 20(R), NOM 100(G), HI 1000(R)
   10  TRANSMIT = RAW,0,6
   12  DE.INE PSU PXAB,PXBA,ALT
   13  STATES = FL 0(R), MAR 10(Y), NOM 1100(G), HM 1500(Y), MI 2000(R)
   14  TRANSMIT = CON,0,9
```

FROM ADDRESS

TO ADDRESS

FORMAT OF TRANSMISSION (RAW COUNTS OR CONVERT TO ENGINEERING UNITS)

```
PROCEDURE
BEGIN, ENTRY
   24  X ← X+1
   26  INCR ← INCR * X
   30  IF (INCR < 100) THEN GO TO 24
   40  XAI ← INCR + 2 * FXD
   41  PXAB ← 1.35 * X
   45  WAIT FOR 500 MS
   50  EXIT
```

STATEMENTS 40 AND 41 WILL CAUSE DATA TO BE TRANSMITTED VIA THE COMM. BUS FROM BUS ADDRESS 6 AND 9 RESPECTIVELY TO BUS ADDRESS 0.

ASD 575

TABLE 1 - AVL SYSTEM HARDWARE ELEMENTS

NOMENCLATURE	ASSEMBLY NO.
AVL SIMULATOR (AVLS)	63D70851G1
DISPLAY & CONTROL MODULE	63E905321G1
ACQUISITION & COMMAND MODULE	63E90540G2
DECOMMUTATOR	075-078G1
PERIPHERAL CABINET	63905477G1
CDC MAGNETIC TAPE UNIT (606)	4620040
CDC MAGNETIC TAPE UNIT (606)	4620040
CDC MAGNETIC TAPE UNIT (606)	4620040
CARD READER (M1000)	63C505582G1
CARD PUNCH (1442)	1442
LINE PRINTER (1612)	25138101
AMPEX MAGNETIC TAPE UNIT (FR1400)	63C505584G1
EXTENDED OPERATORS PANEL	059-523G1
CARD READER SWITCH	63C505997G1
DISPLAY CABINET (PART OF AVLS)	63E90543G1
COMMUNICATIONS CONTROLLER DRAWER (PART OF AVLS)	6390544G1

TABLE 2

TESL STATEMENT SUMMARY

Following is a list of statements required or permitted in each section. An * to the right indicates that the statement is required in its section.

$SPECIFICATION	$PROCEDURE	$TAPS	$DISPLAY
DEFINE ANALOG	ENTRY *	TAPS*	PAGE
DEFINE DISCRETE	EXECUTE	ENTRY	
DEFINE STIMULUS	TERMINATE	EXECUTE	METER
DEFINE PSEUDO	CALL	TERMINATE	PATCH
DEFINE PSEUDO EVENT	DISPLAY PAGE	CALL	EVENT
DEFINE ALPHA	DISPLAY	DISPLAY	CONN
STATES=	ENABLE	replacement	TEXT
INITIAL=	DISABLE	IF	MESSAGE
CHARS=	EXIT*	WAIT	EU
DOWNLIST	replacement	GO TO	GRAPH
DEFINE TIME	IF	ASK	CLOCK
DEFINE SYNC512	WAIT	RETURN*	
DEFINE SYNC128	GO TO	CONTROL	
TRANSMIT	ASK	WHEN	
DEFINE LDB	CONTROL		
	WHEN		
	ABORT		